Voice Driven Dynamic Generation of Webpages

Sahil Modak
Computer Engineering Dept.
Dwarkadas J. Sanghvi COE
Vile Parle, Mumbai, India
sahilmodak1@gmail.com

Sagar Vikmani
Computer Engineering Dept.
Dwarkadas J. Sanghvi COE
Vile Parle, Mumbai, India
sagar.vikmani@gmail.com

Suril Shah
Computer Engineering Dept.
Dwarkadas J. Sanghvi COE
Vile Parle, Mumbai, India
suril.shah.94@gmail.com

Prof. Lakshmi Kurup
Computer Engineering Dept.
Dwarkadas J. Sanghvi COE
Vile Parle, Mumbai, India
lakshmidkurup@gamil.com

***Abstract*— Designing dynamic webpages is a hectic task for experts and a complicated job for new users. This paper presents a system that provides a natural language interface to automate webpage generation. Natural language processing techniques provides a simple interface to users to handle cumbersome applications. The presented system allows users to interact with ease, conveying his requirements to build webpages in natural spoken language. The system thus helps the users not worry about the coding aspect and spontaneously generates the desired webpages by first fetching the user input in the form of speech, understanding the natural language speech of the user, extracting the necessary data and finally creating the corresponding webpage. This automated system can thus save a lot of time of practically any user to design webpages.**

Keywords*— *Human Computer Interaction; Natural Language Processing; Automatic code generation; Webpage generation; Text understanding; Voice driven systems.

I. INTRODUCTION

With the rapidly growing impact of the Internet as a market place for E-commerce, web-page design is also becoming increasingly important as a means of advertising. A webpage is basically composition of various types of controls used for both displaying as well as fetching the necessary information [2]. Major html tags often used are table, anchor, lists, buttons, and combo and text boxes. The physical organization of these constituents and the written HTML code functioning behind them is quite a time consuming task. Several components are used for various purposes. These components can better serve their purpose, and webpages too can be more interactive and easy to use if they are all designed adhering to a user's needs. Subsequently, the complexity of designing tasks can be better managed if the users themselves are more involved thus making user-centric support tools more desirable.

In view of providing one such tool we put forward a system that benefits from the practicality of natural language processing based interfaces. The presented system does not burden the user with the coding aspect, and all the user needs is an elementary idea for the look of the webpage and knowledge of relevant website components to convey the idea to the system. The system then generates the code automatically interpreting the user's spoken commands. This system has been centered about an English speaking user, as the system takes simple voice commands in English and acts according to the interpretations.

The primary objective of our system is to deliver an automated arrangement for the user so that he may himself communicate his requirements and get a corresponding generated output of webpages. The system assists the novel users who do not have the technical expertise in development of webpages.

II. PROBLEM STATEMENT

Websites have fast become the backbone of any business, education or a social organization. Web designing and development have thus seen a comprehensive rise in the technological arena. While, web development is on all-time high, highly skillful and expert personnel are required to design the webpages. And conventional styles of designing webpages have not only proven to be tedious and time consuming, but also have been known to be prone to human induced programming errors. Also, a website can be really unproductive and futile if the user requirements are not satisfactorily met. Furthermore a few existing solutions [4], have had a rather unclear implementation approach, besides the fact that they are limited to build only web forms and need the entire requirement to be typed at once making it more tiresome.

III. OUR APPROACH

In order to cross the aforementioned barriers, the presented solution aims to put forward a system that dynamically builds webpages driven by a user's linguistic commands. To generate the pages, a particular user only has to provide his specific necessities in simple English dialogues. This system interprets them as commands, converts them into text, comprehends the obtained text, digs out the required information and then eventually generates the code automatically. Thus, the automated system eliminates the need of manual coding by field experts and related mistakes. Besides, the webpages can be more freely designed by any individual with a basic idea of the layout and structure of the webpage in accordance with his requirements.

IV. DESIGN AND METHODOLOGY

Researching all the available tools and techniques to carry out the various stages of the project, we have employed the methodology in our solution.

A. *Speech Acquisition:* The user speech is to be considered as commands for the system. These commands would be captured via a microphone enabled computer machine. As the user starts speaking, the speech would be simultaneously acquired by the system.

B. *Text conversion:* The speech of the user captured by the system is simultaneously converted to text for faster and better processing. Due to poor accuracy from offline speech recognition models, the system uses an online Google Speech API [5]. Also, the text converted phrase is displayed to the user to ensure accurate conversion, in case of anomalies the user can either edit the text or speak again.

C. *Text processing:* The text converted from speech is then pre-processed by a tagger which has been incorporated a specifically designed word tagging system. In here, each and every word in the given phrase is associated with a tag similar to a 'Parts-of-speech' tag, which is fetched from a predefined dictionary.

D. *Knowledge Extraction:* This step extracts the important data from the text. This is the basic understanding module where the system interprets the requirements of the user from the processed text. Searching for specific tags in the given command enables our program to extract knowledge efficiently.

E. *Code generation:* This step uses the extracted information as an input. The extracted knowledge is stored in several data structures. The data in this data structures is then used to generate HTML/CSS codes. Once the user commands, the generated code is then saved with an '.html' extension with the styling implemented as inline/internal CSS. The user has the freedom to see the code files and make manual changes to the code.

After successfully saving, the file would a browser preview of the design is displayed to the user to view the results, based on which the user could suggest further editing.

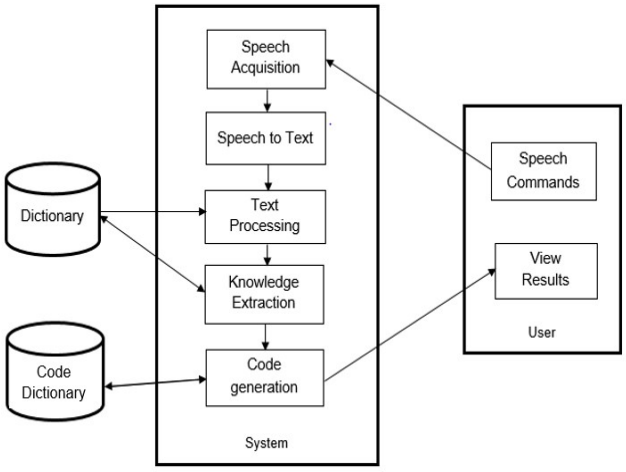

Fig. 1. Example of a figure caption

V. EXAMPLE

Now, consider the following command to understand the architecture better:

A. *Speech Acquisition:* Consider a user saying the phrase: "Draw a table with 2 rows and 3 columns"

B. *Text conversion:* In this step, the user will basically speak the above command in his natural language and the output of this step will be the same command in the form of text.

C. *Text pre-processing:* The input given to this stage will be the output generated from the previous text conversion stage. The output generated by this stage will be of the following form:

Draw → **VB** a → **DT** table → **Noun** with → **Prep**
2 → **Number** rows → **NNS** and → **CC** 3 → **Number**
Columns → **NNS**
Where, VB – Verb
a – Determiner
Prep – Preposition
NNS – Noun, plural
CC – Connector

D. *Knowledge extraction:* The input given to this stage will be the output generated from the previous text pre-processing stage. After receiving this input, this phase is responsible for searching for specific tags. **Verb → what action to perform, Noun → what component to draw, etc**. which enables the identification of the necessary action to perform.

E. *Code Generation*: After identifying the action that needs to be performed, the necessary HTML/CSS code is generated depending on the voice input. This code generation is done using file handling techniques in java.

VI. RESULTS

The implementation results have been very promising. The speech acquisition and text conversion depends very much on the surrounding environment, the background noise, the user's accent and also depends on the microphone's quality of the machine. In a peaceful surrounding and with a decent microphone the text conversion gives an accuracy rate of about 87%. Once, text is generated correctly, the GUI displays a true browser preview of the webpage. Also, the number of tags at this stage have been limited. At the same time, the bigger our dictionary, the more flexible our system would prove to be. The screenshots of the GUI have been provided.

VII. FUTURE SCOPE

With an enormous future scope, the proposed scheme provides vast researching avenues not only in the field of Natural Language Processing but also in the fields of Human Computer Interaction, Artificial Intelligence and Automated Code Generation. Future scope in this domain would be to develop a software that can predict and suggest templates more accurately using machine learning and neural networking. Also, a language independent system could be a further development prospect. More efficient natural language processing can be

introduced with the help of neural networks develop a more intelligent and reliable system. Bringing in more dynamicity with more languages like PHP, JavaScript, etc., different database models and various frameworks like ASP.Net. This software can be used in institutions like schools where deaf or handicapped users can learn to build websites. Besides, currently the system is being developed as a desktop application, further versions can be developed to make the system web based and completely mobile.

VIII. CONCLUSION

A natural language processing based system for dynamically building websites is presented in this document. The system would be driven by users' speech, which will be treated as commands to the system. The speech after having converted to text, will be pre-processed and then processed to extract information from the speech. On the basis of the knowledge interpreted, code will be generated to build static websites centered on HTML and inline CSS. Providing predictive templates would help reduce time and effort consumption. The implemented system so far has managed to achieve low computational complexity and very high accuracy for a limited and essential set of HTML tags. The system is able to provide a smooth interface for user to interact with the computer to build static but functional webpages.

REFERENCES

[1] Imran S. Bajwa, M. Asif Naeem, Riaz-Ul-Amin, M A. Choudhary, Speech Language Processing Interface for Object-Oriented Application Design using a Rulebased Framework, 4th International Conference on Computer Applications 2006 Rangoon, Myanmar

[2] Malaisé Véronique, Zweigenbaum Pierre, Bachimont Bruno, Mining Defining Contexts to Help Structuring Differential Ontologies Terminology, 11:1, 2005

[3] Bajwa, Imran Sarwar, Waqar Aslam, and Syed Irfan Hyder. "Speech Language Engineering System for Automatic Generation of Web based User Forms."

[4] "J.A.R.V.I.S.(Java-Speech-API) ": https://github.com/lkuza2/javaspeech-api#jarvis-java-speech-api

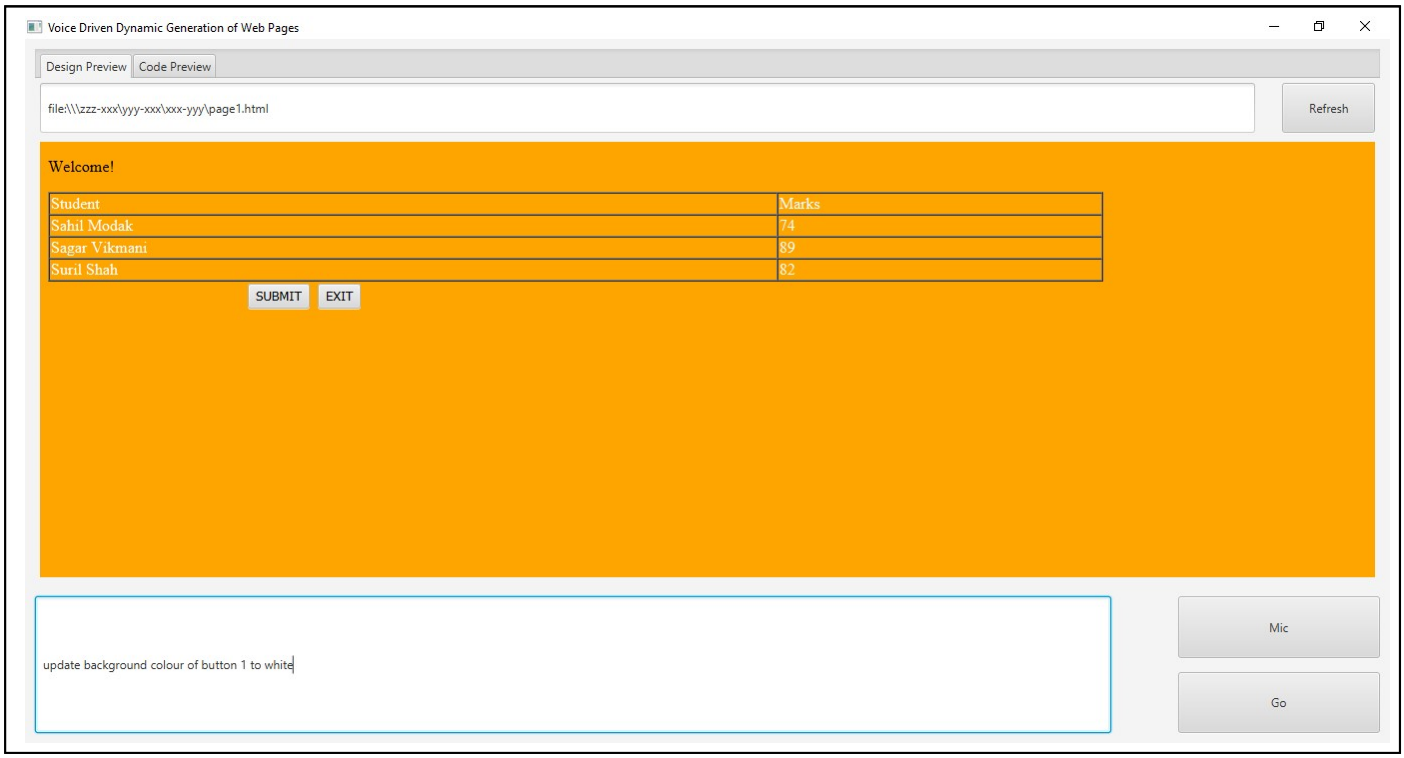

Fig. 2. Screenshot of the graphical user interface, showing an example design preview

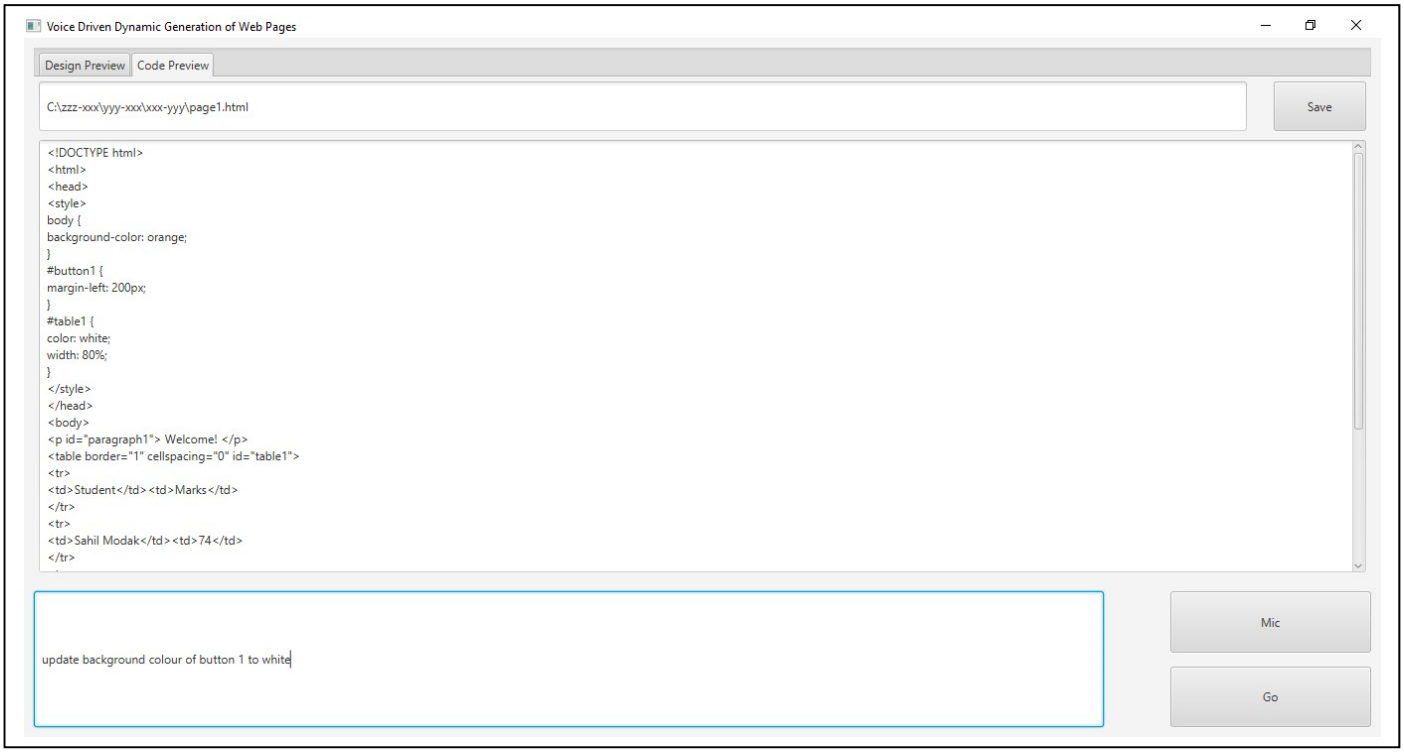

Fig. 3. Screenshot of the graphical user interface, showing an example code preview